PE JC42 MAR 15 2004 PATENT & TRADEMARK OFFICE

#31 LDJ 3-24-04

RECEIVED MAR 22 2004 Technology Center 2100

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Lita** § Group Art Unit: **2155**
§
Serial No.: **09/282,692** § Examiner: **Nguyen, T.**
§
Filed: **03/31/1999** § Atty. Docket No.: **AT9-98-700**
§

For: **Method and System for Using Virtual URLs for Load Balancing (as amended)**

Certificate of Mailing by "Express Mail" Under 37 C.F.R. § 1.10

Express Mail label number: EU867380979US
Date of Mailing: March 15, 2004

I hereby certify that this correspondence is being deposited with the United States Postal Service as "Express Mail Post Office to Addressee" as provided by 37 C.F.R. § 1.10 on the date indicated above with the "Express Mail" mailing label number indicated above in an envelope addressed to:
Mail Stop Appeal Brief--Patent
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450.

By: _____
Joseph R. Burwell, Reg. No 44,468

APPELLANT'S BRIEF IN RESPONSE TO OFFICE ACTION UNDER 37 C.F.R. § 1.192

This brief is filed in triplicate in support of the previously filed Notice of Appeal, which was filed 10/14/2003, and which appealed from the decision of the examiner dated 07/14/2003 rejecting claims 1-22. The fee required under 37 C.F.R. § 1.17(c) for filing a brief in support of an appeal is provided in the Transmittal of Appeal Brief filed herewith.

03/18/2004 CNGUYEN 00000115 501888 09282692
01 FC:1402 330.00 DA

1. REAL PARTY IN INTEREST

The real party in interest in this appeal is International Business Machines Corporation (IBM).

2. RELATED APPEALS AND INTERFERENCES

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

3. STATUS OF CLAIMS

Claims 1-22 are pending in this application; claims 1-22 have been finally rejected; claims 1-22 have been appealed. No claims have been canceled, withdrawn, or allowed.

4. STATUS OF AMENDMENTS

No after-final amendments have been filed.

5. SUMMARY OF INVENTION

The present invention provides a method for distributing client requests across a pool of servers on a per-session basis rather than on a per-connection basis. A server in the pool of servers is allocated a number of sessions such that a client's HTTP connection requests are handled by the same server throughout a user session. A load balancing routine across a set of servers ensures that connection requests from a client during its session are serviced by the same server.

In response to a connection request, a front-end managing server intercepts the request and recognizes that the request will initiate a user session. The managing server acts as a redirector for connection requests; the managing server may query a load balancing routine to determine which server in the pool of servers should service the new session. A unique session identifier is associated with a given server in the pool of servers, and the session identifier is then incorporated into a base URL for the assigned server, thereby forming a "virtual URL" that is returned in an appropriate redirection response to the client, which then automatically issues a new HTTP connection request using the newly generated virtual URL. All subsequent data to the client will incorporate the virtual URL such that subsequent requests from the client will contain the session identifier as part of the URL. Client requests are then routed to the appropriate server in accordance with the URL, and the server can use the session identifier to associate requests with a user session.

6. ISSUES

The issues on appeal are:

whether claims 1-22 under 35 U.S.C. § 102(e) are anticipated by Cherkasova et al., "Hybrid and predictive admission control strategies for a server", U.S. Patent Number 6,360,270, filed 11/16/1998, issued 03/29/2002.

7. GROUPING OF CLAIMS

The claims stand and fall in the following groups:

Group A -- claims 1 and 4-7;

Group B -- claims 2, 3, and 22;

Group C -- claims 8, 9, 13-15, 18, and 21; and

Group D -- claims 10-12, 16, 17, 19, and 20.

8. ARGUMENTS

8.A. -- Was 35 U.S.C. § 102(e) properly applied in a rejection of claims 1 and 4-7 (Group A) as being anticipated over Cherkasova et al.?

Arguments in support of separate patentability

With respect to the grouping of claims, independent claim 1 and its dependent claims 4-7 have been placed in Group A. Independent 1 is the broadest claim in the patent application. Hence, for purposes of this argument, Appellant argues for the patentability of the present invention using claim 1 as an exemplary claim.

Initial review of the teachings of *Cherkasova et al.*

In the background section, Cherkasova et al. briefly explains that the system that is disclosed within Cherkasova et al. is directed to a solution for alleviating quality-of-service problems that are experienced by clients from servers, which is complicated by the fact that most communication over the Internet is performed through the use of so-called stateless protocols. Two general categories of solutions to the quality-of-service problem are mentioned: the addition of processing capacity, and the implementation of "admission control" policies in which only a certain set of client messages are admitted to a server or a set of servers for further processing while the remainder of the messages are refused. Cherkasova et al. presents an admission-control type of solution with the goal of responding in some manner to all incoming messages, whether or not those messages are admitted for further processing. In addition, the system attempts to provide reliable service by admitting messages for on-going sessions, thereby providing those sessions with a high level of quality-of-service.

In its summary section at column 2, lines 56-67, Cherkasova et al. briefly explains the system that is disclosed within Cherkasova et al. in the following manner:

> An admission control system for a server is disclosed including an admission controller that receives a stream of messages from one or more clients targeted for the server. The admission controller relays to the server the messages in the stream that correspond to a number of sessions already underway between the clients and the server. The admission controller also relays to the server the messages in the stream that do not

correspond to sessions already underway if a hybrid and predictive admission control strategy using information provided by a resource monitor indicates that additional sessions can be handled by the server. The admission controller defers the messages otherwise.

The admission controller in this system processes incoming messages based upon whether there are sufficient resources for processing the message and based upon whether the incoming messages correspond to sessions that are already underway at a server. A deferral manager handles the unaccepted messages which were blocked by the admission controller as described in column 4, lines 52-58:

> In one embodiment, the deferral manager transfers the unaccepted messages as a stream of deferred messages 30 to another server (not shown) that replicates the functionality of server 12. For example, if the server is a web server then the deferral manager redirects the deferred messages to another web server, often called a mirror site, that performs the same function as the web server 12.

Cherkasova et al. also discloses that an admission controller can be implemented at a gateway as disclosed at column 10, lines 24-37:

> The gateway 62 is augmented with software elements that provide the functionality of the admission controller 14, the resource monitor 16, and the deferral manager 18. The resource monitor 16 in the gateway 62 monitors the resources of both of the web servers 56 and 58 via the local network 60. The admission controller 14 in the gateway 62 receives arriving messages targeted for the web servers 56 and 58 from the web browsers 44, 46, and 48. The admission controller 14 in the gateway 62 relays the arriving messages that correspond to sessions already underway onto the appropriate one of the web servers 56 and 58 if the resource monitor 16 indicates that sufficient resources are available in the

appropriate web server 56 and 58 to adequately handle additional sessions.

Cherkasova et al. also discloses that an admission controller can be implemented at a proxy server as disclosed at column 10, line 59, to column 11, line 11:

> The proxy server 64 maintains a transaction list 26 that identifies which of the computer systems 68, 70, and 72 have sessions underway with a destination on the network 66. In one embodiment, the transaction list 26 in the proxy server 64 records network addresses on the local network 74 for the computer systems 68, 70, and 72.
>
> The proxy server 64 also contains a resource monitor 16 for monitoring the CPU and storage subsystem utilization in the proxy server, the network utilization in the proxy server, and the network utilization on both the network 66 side and the local network 74 side. The proxy server also contains an admission controller 14 that passes request messages from the computer systems 68, 70, and 72 onto the network 66 if the client request messages correspond to sessions identified in the transaction list 26 of the proxy server. In addition, the admission controller 14 in the proxy server passes client request messages from computer systems 68, 70, and 72 not identified in the transaction list 26 if the resource monitor 16 in the proxy server indicate that sufficient resources are available to allow another session to be established.

Contrasting the present invention and *Cherkasova et al.*

Many important distinctions can be made between the system of the present invention and the system described by Cherkasova et al.. First, in the present invention, the front-end managing server and the servers in the pool of servers may participate in session management. Session identifiers may originate at a server in the pool of servers, and the session identifier may be forwarded to the front-end

managing server for recordation and subsequent use in a redirection response to the client. In contrast, the admission controller in the system taught by Cherkasova et al. performs its own session management without assistance from any of the servers in a cluster of servers; if the admission controller recognizes that a client already has a session as identified by the admission controller, then the admission controller admits the message from the client.

Second, in the present invention, client requests are redirected from the front-end managing server back through the client to a server in the pool of servers. In contrast, if the admission controller in the system taught by Cherkasova et al. determines that it will admit an incoming message, then the message passes through the admission controller to a server; the message is not redirected to a server if a determination is made to perform further processing on the message. In the portion of Cherkasova et al. that states that a message is redirected to a web server, this redirection is explained as only occurring if the admission controller has rejected further processing of the message and shunted the message over to a deferral manager. Hence, even though Cherkasova et al. mentions the use of redirection, it does not use redirection for messages that are being further processed. Moreover, Cherkasova et al. also discloses that any message that is received for an on-going session is admitted and is not deferred; in other words, Cherkasova et al. discloses that the association of a session identifier with a client results in the admission of the messages from that client. Therefore, Cherkasova et al. discloses that no redirected messages would

have a session identifier; otherwise, the message would have been admitted. This is in direct contradiction to the manner in which the present invention operates by associating a session identifier with every redirected message and by redirecting every message that will be processed to a server in the pool of servers.

Third, the present invention ensures that the same server in the pool of servers receives all of the client requests for a particular user session. In contrast, in the system taught by Cherkasova et al., a client request within a user session may be received at any of the servers in the cluster of servers; there is no disclosure that all of the incoming messages from a given client are always admitted to be processed by the same server, and there is no disclosure that the admission controller maintains any information whatsoever to ensure that all of the incoming messages for a given session always go to the same server. This is a consequence of the fact that Cherkasova et al. only discloses that the admission controller performs session management, as described above.

Contrasting independent claim 1 with *Cherkasova et al.*

Given that independent claim 1 is the broadest claim, it is useful to quickly and generally compare the elements of claim 1 against the system of Cherkasova et al. without reference to the rejection. Independent claim 1 reads:

> 1. A method for managing connection requests to a pool of servers identified by a given URL, comprising the steps of:

in response to a connection request from a given client machine that initiates a session, associating a session identifier with a given server in the pool;

using the session identifier in a redirection response;

returning the redirection response to the given client to redirect the connection request to the given server; and

during the session, receiving at the given server any additional connection requests from the given client machine.

With respect to the first element, i.e. "in response to a connection request from a given client machine that initiates a session, associating a session identifier with a given server in the pool", Cherkasova et al. does not disclose that a session identifier is associated with a server in a set of servers. As mentioned above, in the system of Cherkasova et al., there is no coordinated session management between the admission controller, e.g., as embedded in a gateway or a proxy server in front of a set of servers, and the set of servers; the admission controller maintains its own session identifiers.

With respect to the second and third elements, i.e. "using the session identifier in a redirection response" and "returning the redirection response to the given client to redirect the connection request to the given server", as explained above, if the system of Cherkasova et al. redirects an incoming message, it is because the admission controller has determined that the incoming message is not associated with an active session and that the system does not have sufficient resources to service to the incoming message.

With respect to the fourth element, i.e. "during the session, receiving at the given server any additional connection requests from the given client machine", as a consequence of the manner in which the system of Cherkasova et al. performs its session management, the admission controller does not ensure that all incoming messages for a given client are sent to the same server, as mentioned above.

Analyzing the rejection in view of *Cherkasova et al.*

With respect to independent claim 1, the rejection states that the first element of claim 1, i.e. "in response to a connection request from a given client machine that initiates a session, associating a session identifier with a given server in the pool", is disclosed in Cherkasova et al. at column 4, lines 15-35, which reads:

> The admission controller 14 receives the stream of arriving messages 20 which are targeted for the server 12. Each of the arriving messages specifies a client request for the server. Each client request implies an action to be taken by the server in accordance with the predetermined communication protocol which the server processes.
>
> The admission controller 14 processes individual ones of the arriving messages 20 based upon the indications provided by the resource monitor 16 and a determination of whether the arriving messages correspond to sessions already underway with the server 12. In one embodiment, a transaction list 26 identifies any session underway between the server and a requesting client. The admission controller compares client source indications contained in the arriving messages to entries in the transaction list to determine whether the arriving messages correspond to sessions underway. In another embodiment, the admission controller determines whether the arriving messages correspond to sessions underway by

determining whether valid transaction identifiers are contained in the arriving messages.

There are multiple embodiments of a system in Cherkasova et al.. The first embodiment is an admission controller embedded in a single server, whereas the second and third embodiments of the system comprise an admission controller embedded in a gateway and in a proxy server, respectively. At first, Cherkasova et al. describes the admission controller in detail in the first embodiment and then describes the manner in which the admission controller could be used in the second and third embodiments. However, the preambles in each of the independent claims in the present application mention a pool of servers, and the pool of servers is later referenced in the body of each independent claim. Hence, it should be assumed that the second and third embodiments of Cherkasova et al. are the most relevant to the present invention, but there is a need to refer to the detail of the admission controller as described with respect to the first embodiment.

In the portion of Cherkasova et al. above that is cited by the rejection against the first element of claim 1, the "transaction identifiers" are managed by the admission controller without regard to whether the admission controller is embedded in a single server system, a gateway, or a proxy server. Hence, the only server that is associated with a session identifier is the server in which the admission controller is embedded; the admission controller does manage session identifiers for any other servers. Therefore, an argument can be made that a session identifier is associated with a server in a set of servers if a gateway or a proxy

server is included in the set of servers. In other words, the admission controller manages session identifiers, so the session identifiers are associated with a gateway or a proxy server in which the admission controller is embedded. However, once one understands the manner in which the admission controller in the system of Cherkasova et al. manages session identifiers, it should be clear that the system of Cherkasova et al. cannot include the other claimed features of the present invention.

The rejection continues by stating that the second element of claim 1, i.e. "using the session identifier in a redirection response", is disclosed in Cherkasova et al. at column 4, line 50, to column 5, line 8, and at column 6, lines 1-8:

> The deferral manager 18 handles the unaccepted messages 24 which were blocked by the admission controller 14. In one embodiment, the deferral manager transfers the unaccepted messages as a stream of deferred messages 30 to another server (not shown) that replicates the functionality of the server 12. For example, if the server is a web server then the deferral manager redirects the deferred messages to another web server, often called a mirror site, that performs the same function as the web server 12.
>
> In another embodiment wherein the server 12 is a web server, the deferral manager 18 transfers response messages back to the requesting web clients which indicate that a bonus or incentive is available if the deferred request is retried at a later time. For example, if the web server provides a sales transaction to requesting web clients, then the deferred messages 30 are targeted for the deferred requesting clients and may contain encoded information that provides the client with a discount on a later purchase.
>
> In another embodiment, the deferral manager 18 directs the deferred messages 30 to another server that enables the deferred web client to reserve a future time

> interval for access to the server 12. Alternatively, the server may provide a function that enables the deferred web client to reserve a future time. In addition, the deferral manager may transfer a response message to the deferred client that indicates that the request is being deferred.
> ...
> In one embodiment at block 40, the admission controller 14 creates a new entry in the transaction list 26 and writes the IP address of the new request message into the new entry of the transaction list. In another embodiment, the admission controller creates a new entry and writes a new transaction identifier into the new entry of the transaction list 26. The new transaction identifier may be returned to the requesting client that originated the request message as a "cookie" or may be returned to the requesting client in a hidden field of an HTTP form.

The paragraphs about the bonus incentive for a retried request or a reserved future interval are irrelevant. The paragraph about the creation of a new transaction identifier merely describes the details by which a transaction identifier is tracked by the admission controller. Hence, only the first paragraph mentions a redirected message.

However, a redirected message is a message that has been blocked from further processing by the admission controller and is being redirected to another server by a deferral manager. As mentioned at column 4, lines 36-37, "[t]he admission controller 14 accepts the ones of the arriving messages 20 that correspond to sessions underway." Thus, redirected messages do not contain session identifiers in the system of Cherkasova et al..

The rejection continues by stating that the third element of claim 1, i.e. "returning the redirection response to the given client to redirect the connection request to the given

server", is disclosed in Cherkasova et al. at column 4, line 50, to column 5, line 8, and at column 5, line 57, to column 6, line 8, and column 9, line 44, to column 10, line 17. The first and second passages were provided above; the third passage reads:

> The web browsers 44, 46, and 48 transfer HTTP requests via the network 54 and are potential web clients to the web servers 50, 52, 56, and 58. Each HTTP request from the web browsers 44, 46, and 48 contains a Universal Resource Locator (URL), referred to as an "address," that targets one of the web servers 50, 52, 56, and 58. The network 54 routes each HTTP request to either the web server 50 or 52, or the gateway 62, depending on the particular URL contained in the request.
>
> The web server 50 is augmented with software elements that provide functionality of the admission controller 14, the resource monitor 16, and the deferral manager 18. The deferral manager 18 in the web server 50 redirects deferred client request messages to the web server 52. The web server 52 may be a mirror site to the web server 50 or may implement special web server software for handling the deferred client requests as previously described. The resource monitor 16 in the web server 50 may employ the services of an operating system under which it executes to obtain metrics such as CPU, network, or storage subsystem utilization.
>
> In one embodiment, the web server 50 generates transaction identifiers to identify any of the web browsers 44, 46, and 48 to which sessions are underway. The web server 50 may transfer the transaction identifiers to the web browsers 44, 46, and 48 as cookies in response messages to the web browsers. The cookies may be encoded and may have an expiration date and time. The web browsers 44, 46, and 48 include the cookies which they were allocated in subsequent request messages to the web server 50 and the admission controller 14 in subsequent request messages when determining whether to admit the subsequent request messages.
>
> Alternatively, the web server 50 may transfer transaction identifiers to the web browsers 44, 46, and 48 as hidden fields in forms contained in response

> messages to the web browsers. The web browsers submit the forms including hidden transaction identifiers with subsequent request messages to the web server 50 and the admission controller 14 compares the transaction identifiers contained in submitted forms when deciding whether to admit the subsequent request messages.

This passage discusses multiple servers, but only web server 50 contains the functionality of the admission controller. The only other relevant point is that the web server 50 returns transaction identifiers to clients. However, it does not disclose that the transaction identifiers are placed into redirection responses, as required by the claim language of the present application.

The rejection continues by stating that the fourth element of claim 1, i.e. "during the session, receiving at the given server any additional connection requests from the given client machine", is disclosed in Cherkasova et al. at column 5, lines 40-57, which reads:

> Returning to decision block 34, if the new request message does not correspond to a transaction identified in the transaction list 26 then processing proceeds to decision block 36. At decision block 36, the admission controller 14 determines whether sufficient resources are available in the server 12 to adequately service a new session. The determination at decision block 36 is made based upon indications provided by the resource monitor 16 and will be discussed in further detail below. In general, utilization of the resources of the server 12 are measured at regular intervals. If the utilization rises above a specified threshold, then for the next time interval, the admission controller 14 will reject all new sessions and service only existing sessions. Once the utilization falls below the given threshold, then for the next time interval, the admission controller 14 will admit new sessions again while continuing to service existing sessions.

This passage merely describes the manner in which the admission controller operates to always admit messages for on-going sessions at the server in which the admission controller is embedded. This passage does not disclose that additional incoming requests from a particular client are always sent to a particular server in a set of servers after a request has been redirected to the particular server, as required by the claim language.

Rejections are deficient with respect to requirements for a proper anticipation rejection

Clearly, the rejection has not carefully considered the elements of claim 1 nor has the rejection pointed out the claimed features within Cherkasova et al. as is required for a proper anticipation rejection. More importantly, Cherkasova et al. does not disclose the claimed features and cannot be used as an anticipation reference. As stated at MPEP § 2131: "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). "The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). Hence, the rejection of claim 1 over Cherkasova et al. is improper. For this and other reasons, Appellant argues that the position of the Examiner should be reversed and that the rejection of claim 1 should not be upheld.

8.B. -- Was 35 U.S.C. § 102(e) properly applied in a rejection of claims 2, 3, and 22 (Group B) as being anticipated over Cherkasova et al.?

Arguments in support of separate patentability

With respect to the grouping of claims, dependent claims 2 and 3 and independent claim 22 have been placed in Group B. All of these claims are method claims. Claim 22 is similar to claim 1, from which claims 2 and 3 depend; however, claim 22 differs from claim 1 in that claim 22 includes a claim element that specifies the use of a particular form of session identifier, and this claim element is also recited in claims 2 and 3. Hence, for purposes of this argument, Appellant argues for the separate patentability of the claims in Group B using claim 22 as an exemplary claim.

With respect to independent claim 22, the rejection states that Cherkasova et al. discloses the cited features, but the rejection of claim 22 states the following: "generating a virtual URL by modifying the given URL from the connection request to include the session identifier; generating a redirection response comprising the virtual URL ...". In other words, the second element of claim 22 appears to have been ignored by the rejection because no specific citation within Cherkasova et al. was presented by the rejection against this claim element; the rejection possibly relies on the citation of portions of Cherkasova et al. against other claim elements as supposedly disclosing the second element of claim 22. However, Appellant argues that

the rejection of claim 22 is *prima facie* deficient for failing to specifically point out the claimed features within Cherkasova et al. as is required for a proper anticipation rejection.

Turning to the rejection of dependent claim 2, which is one of the claims within Group B along with claim 22, the rejection relies on column 5, line 65, to column 6, line 8, and column 9, line 44, to column 10, line 17, as supposedly disclosing claim 2. However, these portions of Cherkasova et al. have been recited above by Appellant because these portions were also used in the rejection of claim 1, and it is clear that Cherkasova et al. does not disclose the generation of a virtual URL as recited by claim 2, which states: "wherein the step of using the session identifier includes generating a virtual URL". Not only does Cherkasova et al. not disclose the generation of a virtual URL, the system of Cherkasova et al. has no use for a virtual URL given the manner in which the system of Cherkasova et al. operates.

Cherkasova et al. does not disclose the claimed features and cannot be used as an anticipation reference. Hence, the rejection of claim 22 over Cherkasova et al. is improper. For this and other reasons, Appellant argues that the position of the Examiner should be reversed and that the rejection of claim 22 should not be upheld.

8.C. -- Was 35 U.S.C. § 102(e) properly applied in a rejection of claims 8, 9, 13-15, 18, and 21 (Group C) as being anticipated over Cherkasova et al.?

Arguments in support of separate patentability

With respect to the grouping of claims 8, 9, 13-15, 18, and 21 have been placed in Group C. Independent claims 9 and 21 are method claims, whereas independent claim 15 is directed to a computer program product, and independent claim 18 is directed to a server. Claims 15 and 18 correspond in form with claim 9 and may be considered as similar to claim 9 for the purposes of this argument.

Although method claims 1, 9, and 21 are similar, claim 1 is broader than claims 9 and 21. Claim 9 differs from claim 1 in that claim 9 includes a claim element that specifies the use of a load balancing protocol (as does claim 21); claim 8 depends from claim 1 and recites the additional feature of the use of a load balancing protocol. Hence, for purposes of this argument, Appellant argues for the patentability of the claims in Group C using claim 9 as an exemplary claim.

With respect to independent claim 9, the second and third elements of independent claim 9 are substantially similar to the third and fourth elements of independent claim 1. The rejection points to the same portion of Cherkasova et al. that supposedly discloses the features in claim 1. However, as argued by Appellant above with respect to claim 1, Cherkasova et al. does not disclose the features in claim 1.

For the first element of claim 9, the rejection states that Cherkasova et al. discloses "associating a user session

originating from a client machine with a given server in the pool in accordance with a load balancing protocol." However, Cherkasova et al. does not disclose load balancing over a set of servers. Load balancing is a process of attempting to create equal loads on multiple servers within in a set of servers, whereas Cherkasova et al. merely discloses an attempt to prevent overloading on a set of servers as a whole by monitoring the processing load (resources) on the servers as a whole in order to prevent the processing load from exceeding the available resources of the servers as a whole.

Moreover, the rejection states: "It is inherent that admission controller 14, figure 1, does the load balancing job between client and server so as it is clearly use load balancing protocol [sic] to communicate between client and server." This statement is illogical as the concept of load balancing does not apply to a client and a single server.

Cherkasova et al. does not disclose the claimed features and cannot be used as an anticipation reference. Hence, the rejection of claim 9 over Cherkasova et al. is improper. For this and other reasons, Appellant argues that the position of the Examiner should be reversed and that the rejection of claim 9 should not be upheld.

8.D. Was 35 U.S.C. § 102(e) properly applied in a rejection of claims 10-12, 16, 17, 19, and 20 (Group D) as being anticipated over Cherkasova et al.?

Arguments in support of separate patentability

With respect to the grouping of claims 10-12, 16, 17, 19, and 20 have been placed in Group D. Whereas as Group B contains claims that have a claim element that specifies the use of a particular form of session identifier, and whereas Group C contains claims that have a claim element that specifies the use of a load balancing protocol, Group D contains claims that have both additional claim elements. In other words, Group D contains claims that have an additional claim element that specifies a particular form of session identifier and another additional claim element that specifies the use of a load balancing protocol. For purposes of this argument, Appellant argues for the patentability of the present invention using claim 10 as an exemplary claim.

With respect to dependent claim 10, claim 10 recites the features of "generating a virtual URL by modifying a given URL to include a session identifier" and "using the virtual URL to redirect the connection request to the given server". Turning to the rejection of dependent claim 10, the rejection relies on column 5, line 65, to column 6, line 8, and column 9, line 44, to column 10, line 17, as supposedly disclosing claim 10. However, these portions of Cherkasova et al. have been recited above by Appellant because these portions were also used in the rejection of claim 1, and it is clear that Cherkasova et

al. does not disclose the generation of a virtual URL as recited by claim 10. Not only does Cherkasova et al. not disclose the generation of a virtual URL, the system of Cherkasova et al. has no use for a virtual URL given the manner in which the system of Cherkasova et al. operates. In addition, Cherkasova et al. does not disclose load balancing over a set of servers as argued above by Appellant with respect to Group C, including independent claim 9 from which claim 10 depends.

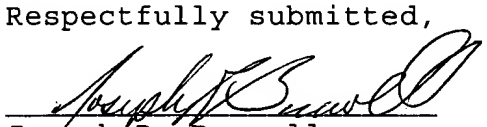
Cherkasova et al. does not disclose the claimed features and cannot be used as an anticipation reference. Hence, the rejection of claim 10 over Cherkasova et al. is improper. For this and other reasons, Appellant argues that the position of the Examiner should be reversed and that the rejection of claim 10 should not be upheld.

9. Conclusion

In view of the above arguments, it is respectfully urged that the rejection of the claims should not be sustained.

DATE: March 15, 2004

Respectfully submitted,



Joseph R. Burwell
Reg. No. 44,468
ATTORNEY FOR APPELLANT

Law Office of Joseph R. Burwell
P.O. Box 28022
Austin, Texas 78755-8022
Voice: 866-728-3688 (866-PATENT8)
Fax: 866-728-3680 (866-PATENT0)
Email: joe@burwell.biz

10. APPENDIX OF CLAIMS

1. A method for managing connection requests to a pool of servers identified by a given URL, comprising the steps of:

in response to a connection request from a given client machine that initiates a session, associating a session identifier with a given server in the pool;

using the session identifier in a redirection response;

returning the redirection response to the given client to redirect the connection request to the given server; and

during the session, receiving at the given server any additional connection requests from the given client machine.

2. The method as described in claim 1 wherein the step of using the session identifier includes generating a virtual URL.

3. The method as described in claim 2 wherein the virtual URL comprises a URL in the connection request modified to include the session identifier.

4. The method as described in claim 1 wherein the session identifier is incorporated in data returned from the given server to the client machine.

5. The method as described in claim 1 further including the step of:

in response to a connection request from the given client machine that terminates the session, inactivating the session identifier.

6. The method as described in claim 1 wherein the given client machine include a browser.

7. The method as described in claim 1 wherein each of the servers in the pool supports a similar set of objects.

8. The method as described in claim 1 wherein the session identifier is associated with a given server as a function of a load balancing protocol.

9. A method for managing connection requests to a pool of servers, comprising the steps of:

responsive to a connection request from a client machine to initiate a user session, associating a user session originating from a client machine with a given server in the pool in accordance with a load balancing protocol;

returning a redirection response to the client machine for the connection request; and

during the user session, receiving at the given server any additional connection requests originating from the client machine.

10. The method as described in claim 9 wherein the associating step comprises:

generating a virtual URL by modifying a given URL to include a session identifier;

using the virtual URL to redirect the connection request to the given server.

11. The method as described in claim 10 further including the step of:

inactivating the virtual URL upon completion of the user session.

12. The method as described in claim 10 wherein all data returned from given server to the client machine includes the session identifier.

13. The method as described in claim 9 wherein each of the servers in the pool supports a similar set of given objects.

14. The method as described in claim 9 wherein each client machine include a Web browser.

15. A computer program product in a computer-readable medium for managing connection requests to a pool of servers, comprising the steps of:

means, responsive to a connection request from a client machine to initiate a user session, for associating a user session originating from a client machine with a given server in the pool in accordance with a load balancing protocol;

means for returning a redirection response to the client machine for the connection request; and

means operative during the user session for receiving at the given server any additional connection requests originating from the client machine.

16. The computer program product as described in claim 15 wherein the associating means comprises:

means for generating a virtual URL by modifying a given URL to include a session identifier;

means for redirecting a given connection request to the given server using the virtual URL.

17. The computer program product as described in claim 16 further including:

means for inactivating the virtual URL upon completion of the user session.

18. A server for managing a pool of servers at a Web site identified by a given URL, comprising:

a processor;

an operating system;

a load balancing routine; and

a redirector routine for managing HTTP connection requests to the Web site, comprising:

means responsive to a connection request from a client machine to initiate a user session for associating a user session originating from a client machine with a given server in the pool in accordance with the load balancing routine;

means for returning a redirection response to the client machine for the connection request; and

means operative during the user session for redirecting to the given server any additional connection requests originating from the client machine.

19. The server as described in claim 18 wherein the means for associating comprises:

means for generating a virtual URL by modifying a given URL to include a session identifier;

means for redirecting a given connection request to the given server using the virtual URL.

20. The server as described in claim 19 wherein the redirector further includes:

means for inactivating the virtual URL upon completion of the user session.

21. A method of managing a pool of servers at a Web site identified by a given URL, comprising the steps of:

responsive to a connection request from a client machine to initiate a user session, associating a user session originating from a client machine with a server in the pool of servers in order to distribute user sessions across the pool of servers in accordance with a load balancing protocol; and

returning a redirection response to a given client machine for the connection request; and

during a given user session initiated from the given client machine, serving content to the given client machine only from its associated server.

22. A method of managing a pool of servers at a Web site identified by a given URL, comprising the steps of:

in response to a connection request containing the given URL from a given client machine that initiates a session, associating a session identifier with a given server in the pool of servers;

generating a virtual URL by modifying the given URL from the connection request to include the session identifier;

generating a redirection response comprising the virtual URL; and

sending the redirection response to the given client machine to redirect the connection request to the given server.